Jay Lee

100357736

CPSC 2221 – 02 Lab 2

1. Query for creating tables

a) Create Tables Query & Foreign Key

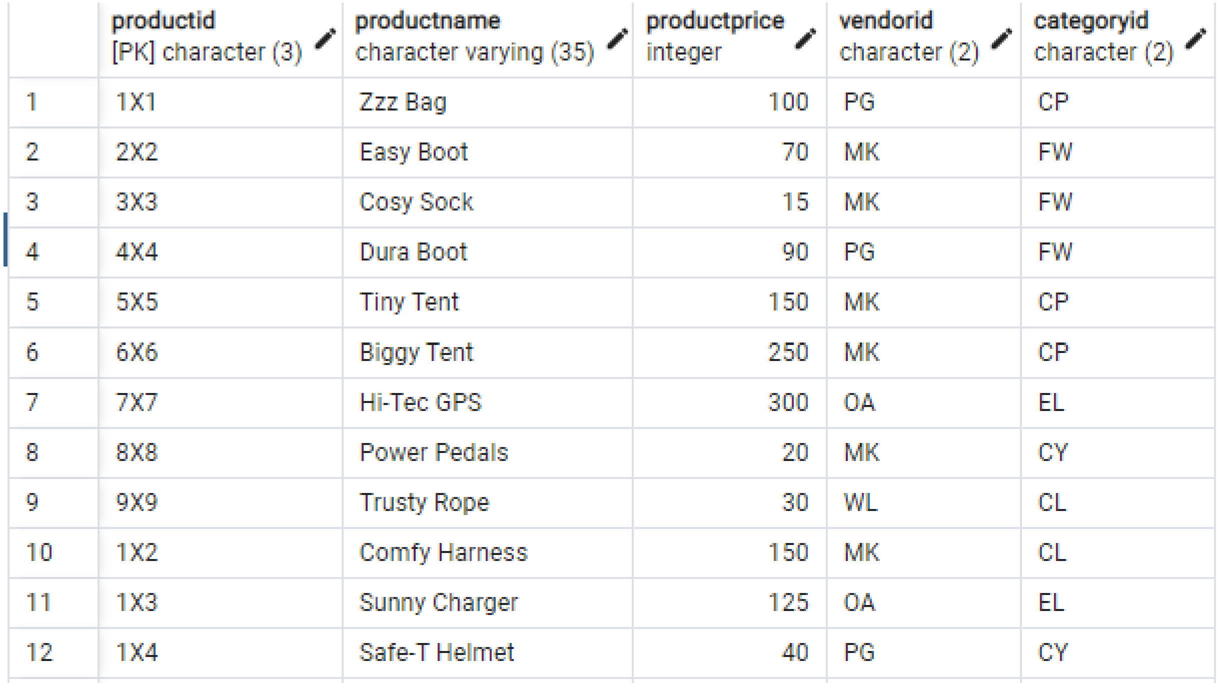
```
DROP TABLE IF EXISTS Region;
CREATE TABLE Region(
        RegionID char(1) PRIMARY KEY,
        RegionName varchar(35) NOT NULL
);

DROP TABLE IF EXISTS Store;
CREATE TABLE Store(
        StoreID varchar(10) PRIMARY KEY,
        StoreZip char(6) NOT NULL,
        RegionID char(1) NOT NULL,
        FOREIGN KEY(RegionID) REFERENCES Region(RegionID)
);

DROP TABLE IF EXISTS Vendor;
CREATE TABLE Vendor(
        VendorID char(2) PRIMARY KEY,
        VendorName varchar(35) NOT NULL
);

DROP TABLE IF EXISTS Category;
```

```
CREATE TABLE Category(
            CategoryID char(2) PRIMARY KEY,
            CategoryName varchar(35) NOT NULL
);

DROP TABLE IF EXISTS Product;
CREATE TABLE Product(
            ProductID char(3) PRIMARY KEY,
            ProductName varchar(35) NOT NULL,
            ProductPrice int NOT NULL,
            VendorID char(2) NOT NULL,
            CategoryID char(2) NOT NULL,
            FOREIGN KEY(VendorID) REFERENCES Vendor(vendorid),
            FOREIGN KEY (CategoryID) REFERENCES Category(CategoryID)
);

DROP TABLE IF EXISTS Customer;
CREATE TABLE Customer(
            CustomerID char(7) PRIMARY KEY,
            CustomerName varchar(35) NOT NULL,
            CustomerZip char(6) NOT NULL
);

DROP TABLE IF EXISTS SalesTransaction;
CREATE TABLE SalesTransaction(
            TID char(4) PRIMARY KEY,
            CustomerID char(7) NOT NULL,
            StoreID varchar(10) NOT NULL,
        TDate date,
        FOREIGN KEY (CustomerID) REFERENCES Customer(CustomerID),
        FOREIGN KEY (StoreID) REFERENCES Store(StoreID)
);

DROP TABLE IF EXISTS Soldvia;
CREATE TABLE Soldvia(
        ProductID char(3) NOT NULL,
        TID char(4) NOT NULL,
        FOREIGN KEY (ProductID) REFERENCES Product(ProductID),
        FOREIGN KEY (TID) REFERENCES SalesTransaction(TID),
        NoOfItems int NOT NULL,
        PRIMARY KEY (ProductID, TID)
);
```

2. Query for creating tables

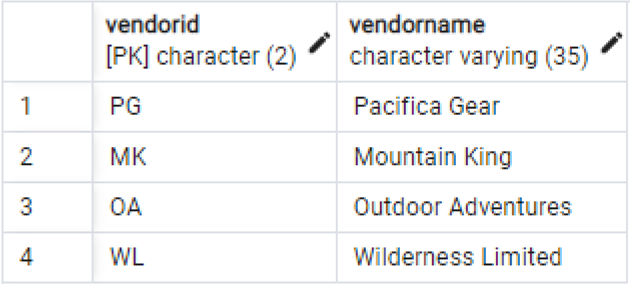
Queries:

1. SELECT *
   FROM Product;



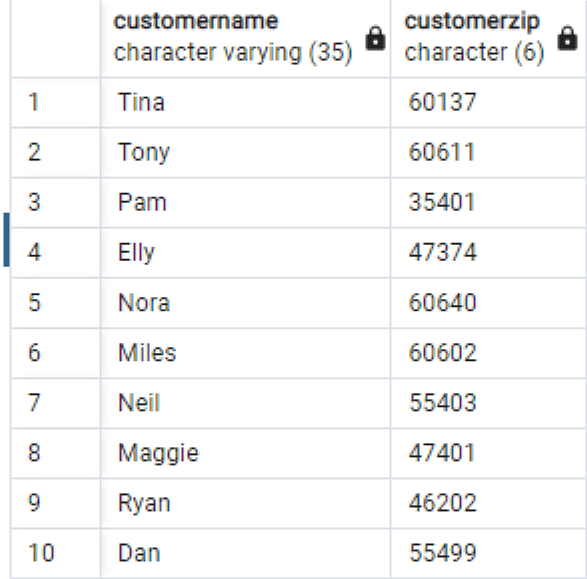
	productid [PK] character (3)	productname character varying (35)	productprice integer	vendorid character (2)	categoryid character (2)
1	1X1	Zzz Bag	100	PG	CP
2	2X2	Easy Boot	70	MK	FW
3	3X3	Cosy Sock	15	MK	FW
4	4X4	Dura Boot	90	PG	FW
5	5X5	Tiny Tent	150	MK	CP
6	6X6	Biggy Tent	250	MK	CP
7	7X7	Hi-Tec GPS	300	OA	EL
8	8X8	Power Pedals	20	MK	CY
9	9X9	Trusty Rope	30	WL	CL
10	1X2	Comfy Harness	150	MK	CL
11	1X3	Sunny Charger	125	OA	EL
12	1X4	Safe-T Helmet	40	PG	CY

2. SELECT VendorID, VendorName
   FROM Vendor;



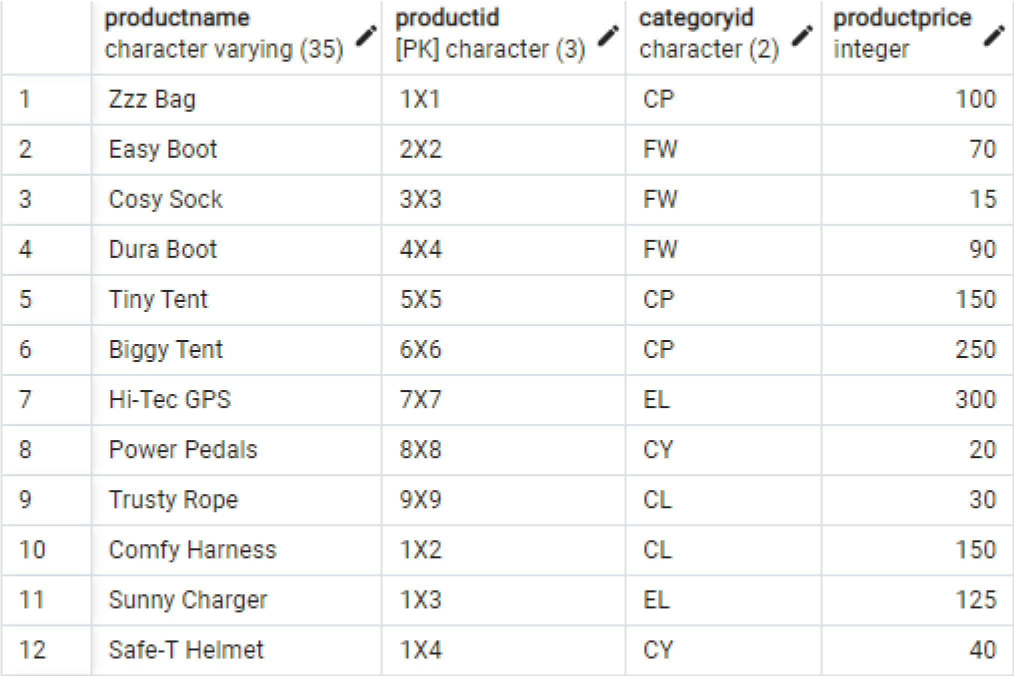
	vendorid [PK] character (2)	vendorname character varying (35)
1	PG	Pacifica Gear
2	MK	Mountain King
3	OA	Outdoor Adventures
4	WL	Wilderness Limited

3. SELECT CustomerName, CustomerZip
   FROM Customer;



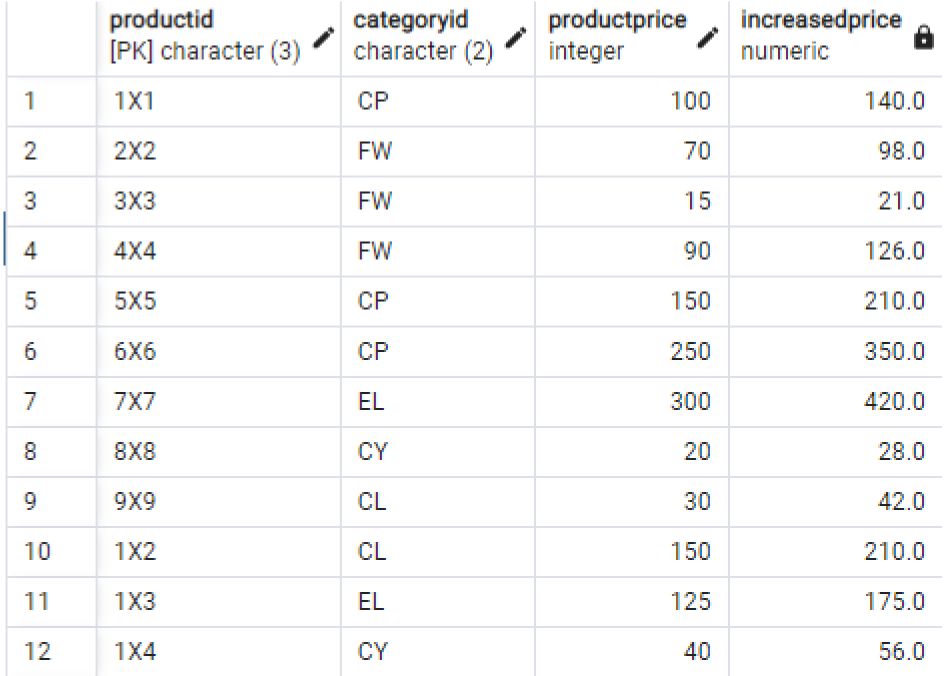
	customername character varying (35)	customerzip character (6)
1	Tina	60137
2	Tony	60611
3	Pam	35401
4	Elly	47374
5	Nora	60640
6	Miles	60602
7	Neil	55403
8	Maggie	47401
9	Ryan	46202
10	Dan	55499

4. SELECT ProductName, ProductID, CategoryID, ProductPrice
   FROM Product;



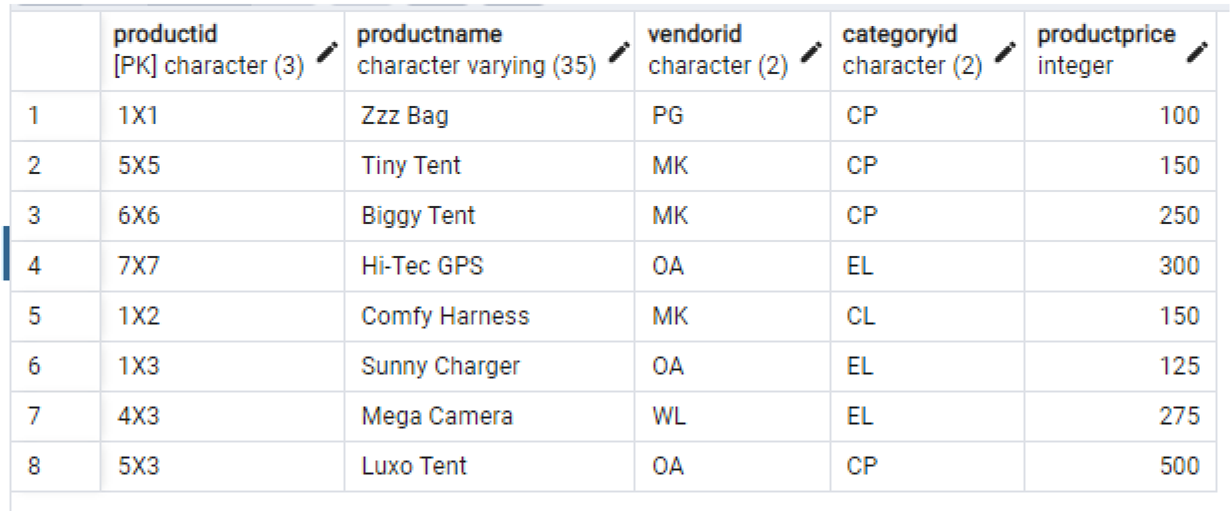
	productname character varying (35)	productid [PK] character (3)	categoryid character (2)	productprice integer
1	Zzz Bag	1X1	CP	100
2	Easy Boot	2X2	FW	70
3	Cosy Sock	3X3	FW	15
4	Dura Boot	4X4	FW	90
5	Tiny Tent	5X5	CP	150
6	Biggy Tent	6X6	CP	250
7	Hi-Tec GPS	7X7	EL	300
8	Power Pedals	8X8	CY	20
9	Trusty Rope	9X9	CL	30
10	Comfy Harness	1X2	CL	150
11	Sunny Charger	1X3	EL	125
12	Safe-T Helmet	1X4	CY	40

5.
```
SELECT ProductID, CategoryID, ProductPrice, ProductPrice * 1.4 AS IncreasedPrice
FROM Product;
```



	productid [PK] character (3)	categoryid character (2)	productprice integer	increasedprice numeric
1	1X1	CP	100	140.0
2	2X2	FW	70	98.0
3	3X3	FW	15	21.0
4	4X4	FW	90	126.0
5	5X5	CP	150	210.0
6	6X6	CP	250	350.0
7	7X7	EL	300	420.0
8	8X8	CY	20	28.0
9	9X9	CL	30	42.0
10	1X2	CL	150	210.0
11	1X3	EL	125	175.0
12	1X4	CY	40	56.0

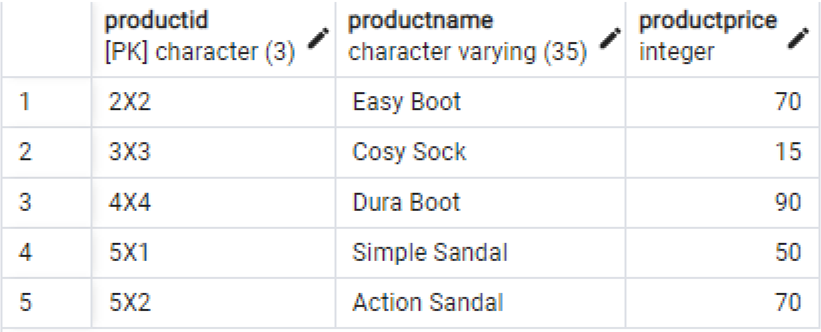
6.
```
SELECT ProductID, ProductName, VendorID, CategoryID, ProductPrice
FROM Product
WHERE ProductPrice >= 100;
```



	productid [PK] character (3)	productname character varying (35)	vendorid character (2)	categoryid character (2)	productprice integer
1	1X1	Zzz Bag	PG	CP	100
2	5X5	Tiny Tent	MK	CP	150
3	6X6	Biggy Tent	MK	CP	250
4	7X7	Hi-Tec GPS	OA	EL	300
5	1X2	Comfy Harness	MK	CL	150
6	1X3	Sunny Charger	OA	EL	125
7	4X3	Mega Camera	WL	EL	275
8	5X3	Luxo Tent	OA	CP	500

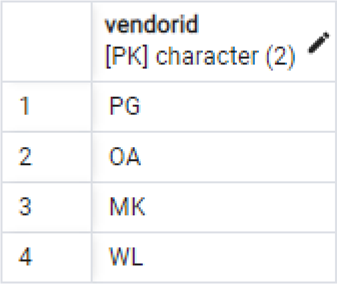
7.
```
SELECT ProductID, ProductName, ProductPrice
FROM Product
```

WHERE CategoryID = 'FW' AND ProductPrice <= 200;



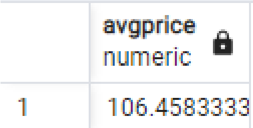
	productid [PK] character (3)	productname character varying (35)	productprice integer
1	2X2	Easy Boot	70
2	3X3	Cosy Sock	15
3	4X4	Dura Boot	90
4	5X1	Simple Sandal	50
5	5X2	Action Sandal	70

8. SELECT VendorID
FROM Vendor
GROUP BY VendorID;



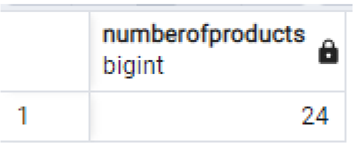
	vendorid [PK] character (2)
1	PG
2	OA
3	MK
4	WL

9. SELECT AVG(ProductPrice) AS AvgPrice
FROM Product;



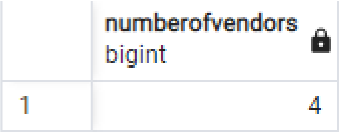
	avgprice numeric
1	106.4583333

10. SELECT count(*) AS NumberOfProducts
FROM Product;



	numberofproducts bigint
1	24

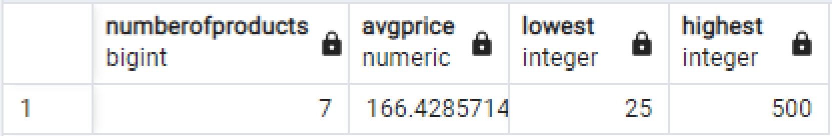
11. SELECT COUNT(DISTINCT VendorID) AS NumberOfVendors
FROM Product;



	numberofvendors bigint
1	4

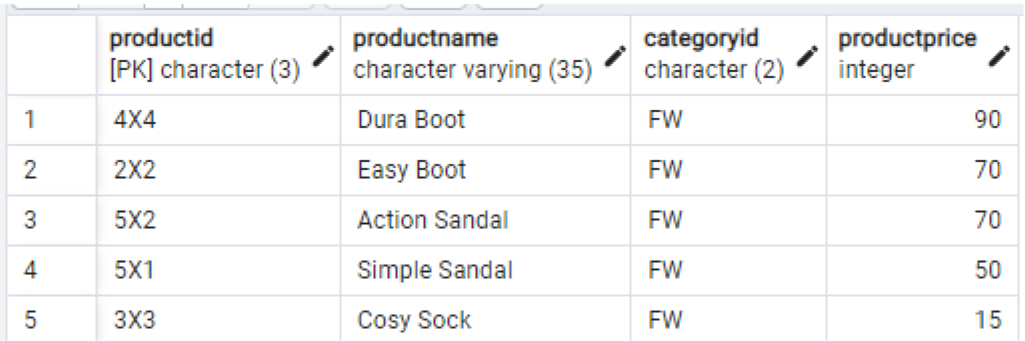
12. SELECT COUNT(*) AS NumberOfProducts, AVG(ProductPrice) AS AVGPrice,
MIN(ProductPrice) AS Lowest, MAX(ProductPrice) AS Highest
FROM Product

```
WHERE CategoryID = 'CP';
```



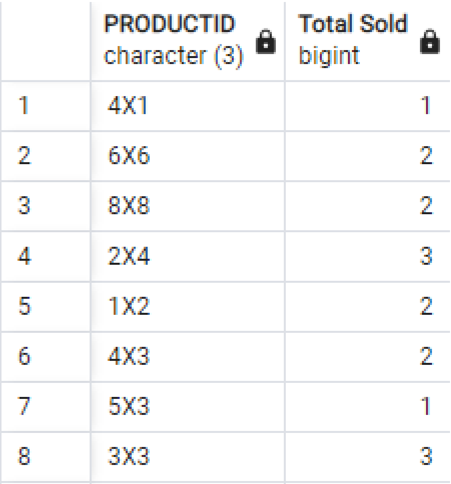
	numberofproducts bigint	avgprice numeric	lowest integer	highest integer
1	7	166.4285714	25	500

```
13. SELECT ProductID, ProductName, CategoryID, ProductPrice
    FROM Product
    WHERE CategoryID = 'FW'
    ORDER BY ProductPrice DESC;
```



	productid [PK] character (3)	productname character varying (35)	categoryid character (2)	productprice integer
1	4X4	Dura Boot	FW	90
2	2X2	Easy Boot	FW	70
3	5X2	Action Sandal	FW	70
4	5X1	Simple Sandal	FW	50
5	3X3	Cosy Sock	FW	15

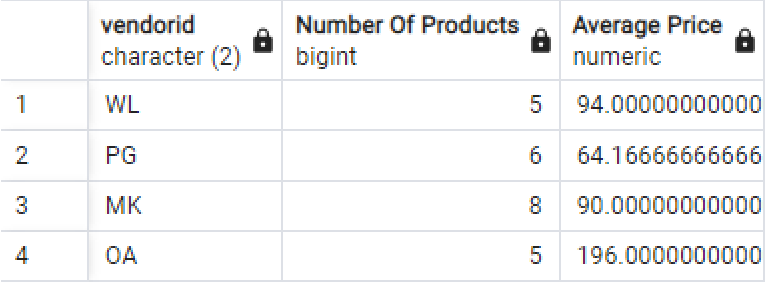
```
14. SELECT ProductID AS "PRODUCTID", COUNT(NoOfItems) AS "Total Sold"
    FROM Soldvia
    GROUP BY ProductId;
```



	PRODUCTID character (3)	Total Sold bigint
1	4X1	1
2	6X6	2
3	8X8	2
4	2X4	3
5	1X2	2
6	4X3	2
7	5X3	1
8	3X3	3

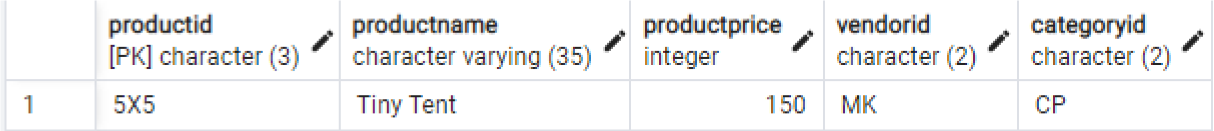
```
15. SELECT VendorID, COUNT(ProductName) AS "Number Of Products", AVG(ProductPrice) AS
    "Average Price"
    FROM Product
```

GROUP BY VendorID;



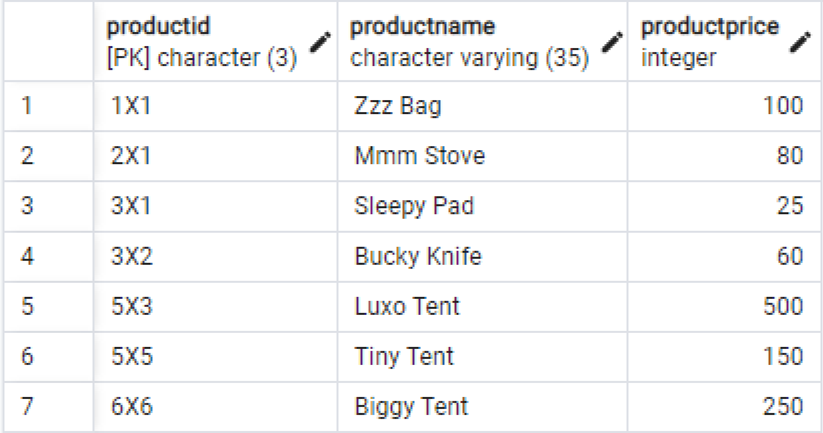
	vendorid character (2)	Number Of Products bigint	Average Price numeric
1	WL	5	94.00000000000
2	PG	6	64.16666666666
3	MK	8	90.00000000000
4	OA	5	196.0000000000

16. SELECT *
    FROM Product
    WHERE ProductName
    LIKE 'Tiny%';



	productid [PK] character (3)	productname character varying (35)	productprice integer	vendorid character (2)	categoryid character (2)
1	5X5	Tiny Tent	150	MK	CP

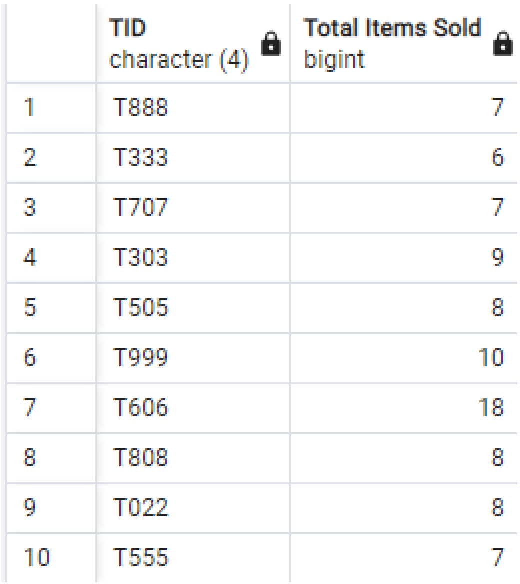
17. SELECT ProductID, ProductName, ProductPrice
    FROM Product
    WHERE CategoryID = 'CP'
    ORDER BY ProductID



	productid [PK] character (3)	productname character varying (35)	productprice integer
1	1X1	Zzz Bag	100
2	2X1	Mmm Stove	80
3	3X1	Sleepy Pad	25
4	3X2	Bucky Knife	60
5	5X3	Luxo Tent	500
6	5X5	Tiny Tent	150
7	6X6	Biggy Tent	250

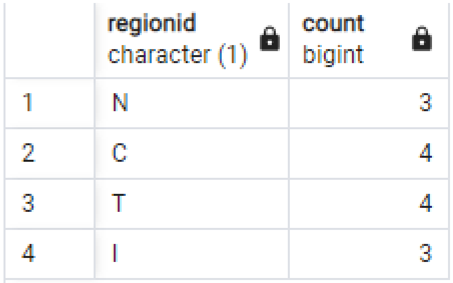
18. SELECT TID AS "TID", SUM(NoOfItems) AS "Total Items Sold"
    FROM SoldVia
    GROUP BY TID

```
HAVING SUM(NoOfItems) > 5;
```



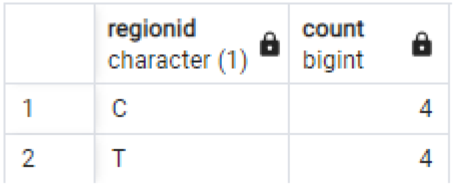
	TID character (4)	Total Items Sold bigint
1	T888	7
2	T333	6
3	T707	7
4	T303	9
5	T505	8
6	T999	10
7	T606	18
8	T808	8
9	T022	8
10	T555	7

19.
```
SELECT RegionID, COUNT(StoreID)
FROM Store
GROUP BY RegionID
```



	regionid character (1)	count bigint
1	N	3
2	C	4
3	T	4
4	I	3

20.
```
SELECT RegionID, COUNT(StoreID)
FROM Store
GROUP BY RegionID
HAVING COUNT(StoreID) >= 4;
```



	regionid character (1)	count bigint
1	C	4
2	T	4